

IN THE CLAIMS

Please amend claims 1, 9 and 17-24 as follows:

1. (CURRENTLY AMENDED) A method of ~~performing~~ simplifying a query in a computer to retrieve data from a database, comprising:

(a) determining whether a query includes a self join that is transitively derived through table expressions having UNION operators; and

(b) simplifying the query to eliminate the table expressions and to reduce the query to an equivalent query over tables, when the query includes the self join that is transitively derived through the table expressions having the UNION operators.

2. (ORIGINAL) The method of claim 1, wherein the determining step (a) and simplifying step (b) are performed for each quantifier in the query, in order to determine whether the quantifier is part of a self-join that can be eliminated.

3. (PREVIOUSLY PRESENTED) The method of claim 2, wherein the simplifying step (b) comprises:

rewriting the table expressions such that quantified columns to be eliminated are substituted by other quantified columns in a same equivalence class; and

once all quantified columns have been rewritten, removing the quantifier from the query.

4. (ORIGINAL) The method of claim 3, wherein the equivalence class is determined by the presence of join predicates establishing equality among quantified columns.

5. (ORIGINAL) The method of claim 3, wherein the quantifier is only considered for elimination when it is represented in each equivalence class.

6. (ORIGINAL) The method of claim 1, wherein the determining step (a) and simplifying step (b) are implemented as a rule-based transformation, the determining step (a) comprises a condition part of the rule, and the simplifying step (b) comprises an action part of the rule that is performed when the condition part is true.

7. (ORIGINAL) The method of claim 6, wherein the condition part of the rule requires:
- (1) that a quantifier  $Q_i$  in the query is a FOR EACH quantifier;
  - (2) that  $Q_i$  is defined within a SELECT operator that implements an inner join;
  - (3) that there are at least two quantifiers  $Q_i$  and  $Q_j$  in the SELECT operator;
  - (4) computing equivalence classes among columns in the SELECT operator, wherein the equivalence classes are divided into three classes:
    - (A) Type 1 equivalence classes are sets of quantified columns over quantifiers other than  $Q_i$ ;
    - (B) Type 2 equivalence classes are sets of quantified columns that reference  $Q_i$  and at least one other quantifier  $Q_j$ ;
    - (C) Type 3 equivalence classes are sets of quantified columns that only reference  $Q_i$ ;
  - (5) for each set  $S_i$  of Type 2 equivalence classes, separating quantified columns according to the quantifiers over which they range;
  - (6) enumerating all combinations of quantified columns  $KEY_1, \dots, KEY_n$  over  $Q_i$ ;
  - (7) retaining only those  $Q_i$  where  $KEY_i$  comprises key columns;
  - (8) computing the derivation of Superset columns from which  $KEY_i$  is derived and adding them to  $KEY_i$ ;
  - (9) for each quantifier  $Q_j$  in the SELECT operator where  $Q_j \neq Q_i$ , if  $Q_j$  has a quantified column in each Type 2 equivalence class:
    - (A) computing the derivation of Subset columns from which  $Q_i$  is derived;
    - (B) for each derivation in  $Q_i$ :
      - (i) for all lists of quantified columns in the derivation:
        - (I) the list of quantified columns matches at least one list of quantified columns in  $KEY_i$ .

8. (ORIGINAL) The method of claim 6, wherein the action part of the rule requires:

- (1) for each Type 2 equivalence class:
  - (a) replacing all occurrences of quantified columns over  $Q_i$  with quantified columns in the same Type 2 equivalence class;
- (2) removing  $Q_i$  from the query.

9. (CURRENTLY AMENDED) A computer-implemented apparatus for performing simplifying a query, comprising:

a computer, wherein the query is performed in the computer to retrieve data from a database;

logic, performed by the computer system, for:

(a) determining whether a query includes a self join that is transitively derived through table expressions having the UNION operators; and

(b) simplifying the query to eliminate the table expressions and to reduce the query to an equivalent query over tables, when the query includes the self join that is transitively derived through the table expressions having the UNION operators.

10. (ORIGINAL) The apparatus of claim 9, wherein the logic for determining (a) and logic for simplifying (b) are performed for each quantifier in the query, in order to determine whether the quantifier is part of a self-join that can be eliminated.

11. (PREVIOUSLY PRESENTED) The apparatus of claim 10, wherein the logic for simplifying (b) comprises logic for:

rewriting the table expressions such that quantified columns to be eliminated are substituted by other quantified columns in a same equivalence class; and

once all quantified columns have been rewritten, removing the quantifier from the query.

12. (ORIGINAL) The apparatus of claim 11, wherein the equivalence class is determined by the presence of join predicates establishing equality among quantified columns.

13. (ORIGINAL) The apparatus of claim 11, wherein the quantifier is only considered for elimination when it is represented in each equivalence class.

14. (ORIGINAL) The apparatus of claim 9, wherein the logic for determining (a) and logic for simplifying (b) are implemented as a rule-based transformation, the logic for determining (a) comprises a condition part of the rule, and the logic for simplifying (b) comprises an action part of the rule that is performed when the condition part is true.

15. (ORIGINAL) The apparatus of claim 14, wherein the condition part of the rule requires:

- (1) that a quantifier  $Q_i$  in the query is a FOR EACH quantifier;
- (2) that  $Q_i$  is defined within a SELECT operator that implements an inner join;
- (3) that there are at least two quantifiers  $Q_i$  and  $Q_j$  in the SELECT operator;
- (4) computing equivalence classes among columns in the SELECT operator, wherein the equivalence classes are divided into three classes:

(A) Type 1 equivalence classes are sets of quantified columns over quantifiers other than  $Q_i$ ;

(B) Type 2 equivalence classes are sets of quantified columns that reference  $Q_i$  and at least one other quantifier  $Q_j$ ;

(C) Type 3 equivalence classes are sets of quantified columns that only reference  $Q_i$ ;

(5) for each set  $S_i$  of Type 2 equivalence classes, separating quantified columns according to the quantifiers over which they range;

(6) enumerating all combinations of quantified columns  $KEY_1, \dots, KEY_n$  over  $Q_i$ ;

(7) retaining only those  $Q_i$  where  $KEY_i$  comprises key columns;

(8) computing the derivation of Superset columns from which  $KEY_i$  is derived and adding them to  $KEY_i$ ;

(9) for each quantifier  $Q_j$  in the SELECT operator where  $Q_j \supset Q_i$ , if  $Q_j$  has a quantified column in each Type 2 equivalence class:

(A) computing the derivation of Subset columns from which  $Q_i$  is derived;

(B) for each derivation in  $Q_i$ :

(i) for all lists of quantified columns in the derivation:

(I) the list of quantified columns matches at least one list of quantified columns in  $KEY_i$ .

16. (ORIGINAL) The apparatus of claim 14, wherein the action part of the rule requires:

(1) for each Type 2 equivalence class:

(a) replacing all occurrences of quantified columns over  $Q_i$  with quantified columns in the same Type 2 equivalence class;

(2) removing  $Q_i$  from the query.

17. (CURRENTLY AMENDED) An ~~article of manufacture~~ device embodying logic for ~~performing~~ simplifying a query in a computer to retrieve data from a database, the logic comprising:

- (a) determining whether a query includes a self join that is transitively derived through table expressions having the UNION operators; and
- (b) simplifying the query to eliminate the table expressions and to reduce the query to an equivalent query over tables, when the query includes the self join that is transitively derived through the table expressions having the UNION operators.

18. (CURRENTLY AMENDED) The ~~article of manufacture~~ device of claim 17, wherein the determining step (a) and simplifying step (b) are performed for each quantifier in the query, in order to determine whether the quantifier is part of a self-join that can be eliminated.

19. (CURRENTLY AMENDED) The ~~article of manufacture~~ device of claim 18, wherein the simplifying step (b) comprises:

- rewriting the table expressions such that quantified columns to be eliminated are substituted by other quantified columns in a same equivalence class; and
- once all quantified columns have been rewritten, removing the quantifier from the query.

20. (CURRENTLY AMENDED) The ~~article of manufacture~~ device of claim 19, wherein the equivalence class is determined by the presence of join predicates establishing equality among quantified columns.

21. (CURRENTLY AMENDED) The ~~article of manufacture~~ device of claim 19, wherein the quantifier is only considered for elimination when it is represented in each equivalence class.

22. (CURRENTLY AMENDED) The ~~article of manufacture~~ device of claim 17, wherein the determining step (a) and simplifying step (b) are implemented as a rule-based transformation, the determining step (a) comprises a condition part of the rule, and the simplifying step (b) comprises an action part of the rule that is performed when the condition part is true.

23. (CURRENTLY AMENDED) The ~~article of manufacture~~ device of claim 22, wherein the condition part of the rule requires:

- (1) that a quantifier  $Q_i$  in the query is a FOR EACH quantifier;
- (2) that  $Q_i$  is defined within a SELECT operator that implements an inner join;
- (3) that there are at least two quantifiers  $Q_i$  and  $Q_j$  in the SELECT operator;
- (4) computing equivalence classes among columns in the SELECT operator, wherein the equivalence classes are divided into three classes:
  - (A) Type 1 equivalence classes are sets of quantified columns over quantifiers other than  $Q_i$ ;
  - (B) Type 2 equivalence classes are sets of quantified columns that reference  $Q_i$  and at least one other quantifier  $Q_j$ ;
  - (C) Type 3 equivalence classes are sets of quantified columns that only reference  $Q_i$ ;
- (5) for each set  $S_i$  of Type 2 equivalence classes, separating quantified columns according to the quantifiers over which they range;
- (6) enumerating all combinations of quantified columns  $KEY_1, \dots, KEY_n$  over  $Q_i$ ;
- (7) retaining only those  $Q_i$  where  $KEY_i$  comprises key columns;
- (8) computing the derivation of Superset columns from which  $KEY_i$  is derived and adding them to  $KEY_i$ ;
- (9) for each quantifier  $Q_j$  in the SELECT operator where  $Q_j \supset Q_i$ , if  $Q_j$  has a quantified column in each Type 2 equivalence class:
  - (A) computing the derivation of Subset columns from which  $Q_i$  is derived;
  - (B) for each derivation in  $Q_i$ :
    - (i) for all lists of quantified columns in the derivation:
      - (I) the list of quantified columns matches at least one list of quantified columns in  $KEY_i$ .

24. (CURRENTLY AMENDED) The ~~article of manufacture~~ device of claim 22, wherein the action part of the rule requires:

- (1) for each Type 2 equivalence class:
  - (a) replacing all occurrences of quantified columns over  $Q_i$  with quantified columns in the same Type 2 equivalence class;
- (2) removing  $Q_i$  from the query.